Implementing a Cocktail-party Processor via Time-frequency Masking

Tao C. Lee and Benedikt Fasel
EPFL

***Abstract*—The ability of human auditory systems to focus on one signal and ignore other signals in an auditory scene where several auditory events are taking place, often referred to as cocktail-party effect, is a key to localization of sound sources. This ability is partly made possible by interaural cues – Interaural Time Differences (ITDs) and Interaural Level Differences (ILDs) – between the input ear signals that assist the estimation of source azimuth angles, and separation of the signal of the desired direction from signals of non-desire directions. In this paper, we investigate simplified techniques to source separation of sound sources based on inter-channel cues. Particular emphasis is put on the selection of time-frequency masks and its effects on the quality of source separation.**

***Index Terms*—cocktail-party processing, source separation.**

I. Introduction

THE cocktail-party effect is the ability of the human auditory system to select one desired sound from a mixture of noise, reflection or other sound sources. The name cocktail party comes from a common scene in a party, where many conversations are taking place simultaneously, humans may focus their attentions to one voice and ignore other voices and noise which are possible equally strong in loudness.

It is not entirely clear how human auditory systems complete this amazing task. However, this ability can be partly understood as the utilization of interaural cues – ITDs and ILDs – between two ear input signals for estimation of sound azimuth angles and separation of sound sources. Based on the ITD and ILD of two ear input signals, the human auditory systems can separate the sound source of the desired direction from sound sources of non-desired directions. Without a doubt, this ability to discern sound from selective directions plays an important role in the localization of sound sources.

The concept of the cocktail-party processor, motivated by simulating electronically the "cocktail party effect," has been proposed by researchers [1, 2]. The early algorithm simulated neural excitation patterns based on specific psychological assumptions about human auditory systems. By modeling the stages of signal processing in auditory system, several key parameters of audio scene such as the azimuth angles can be approximately reconstructed. These parameters are then used to control filters to process the input signals. One practical application of cocktail-party processors can be hearing-aid devices for medical uses [3].

Although early proposals of cocktail-party processors target modeling of the human auditory systems, their complexities are high [1, 2, 3]. Thus, proposals to simplify the complexities of cocktail-party processors have received attention [4]. The simplified approach makes use of Fourier transform based estimation of binaural localization cues (ITD and ILD), and staged enhancement techniques controlled as functions of these binaural cues. Signal processing techniques such as blind source separation and noise reduction have been employed to boost performance [4].

In this paper, we make the observations that blind source separation plays an important role in the performance of the cocktail-party processor. We saw that past researchers [4, 6] have not fully explored the selection of masking schemes and their performance impacts. In particular, we investigate the selection of time-frequency masks and its effects on the quality of source separation.

II. Review of Existing Techniques

The various cocktail-party techniques found in literature can be separated into two classes: The first and older technique consists in estimating the direction of the different sound sources and then uses beam forming to separate them. The second and newer approach tries to construct appropriate time-frequency masks that separate the spectrogram into the different sources. The newer approach using time-frequency masks will be described in more details in the following paragraphs.

Two papers [4, 7] summarize the problem best and provide good descriptions how to build a cocktail-party processor. Both papers intend to separate a mixture of different speech signals where each source is at a different location. The algorithms are designed to work for the degenerate case: there are more sources present than available recordings. For example one has a stereo recording from two microphones separated at a certain distance but three sound sources. As the emphasis was on speech signals an important property of speech is exploited: it is sparse. This means that at a given time and frequency there is only one single source active. [7] defines this property very nicely and calls it the "W-disjoint orthogonality" and they extend the principle to situations where this disjointness is only approximately satisfied. Based on this orthogonality principle it is then theoretically possible to attribute the energy of each time-frequency point to a

Tao. C. Lee and Benedikt Fasel are master students at EPFL.

source. However, in practice things are different and become a bit more complicated. The main difficulty lies in estimating the source statistics that then allow to build such a mask. The most common approach to find the source statistics is to use inter channel time and level differences (ICTD and ICLD). In the ideal case, each source has different and unique ICTD and ICLD. Thus, for each time-frequency point the ICTD and ICLD can be computed and then attributed to the correct source.

[7] proposes a very simple and intuitive way to compute the source parameters: Let x_1 and x_2 be the short time Fourier transforms of the two input channels. The transforms have been obtained by an appropriate windowing function and window length. It is shown in the paper that the optimal window size for a sampling rate of 16kHz is 1024 and that the Hamming window produces best results. From the ratio of x_1 and x_2 the parameters $\alpha[k,l]$ and $\delta[k,l]$ can be computed as follows:

$$R_{21}[k,l] = \frac{x_2[k,l]}{x_1[k,l]} \quad (1)$$

$$a[k,l] = |R_{21}[k,l]| \quad (2)$$

$$\alpha[k,l] = a[k,l] - \frac{1}{a[k,l]} \quad (3)$$

$$\delta[k,l] = -\frac{1}{l\,\omega_0} \quad R_{21}[k,l] \quad (4)$$

Based on these parameters the time-frequency mask can then be found using a clustering function or a maximum likelihood function for example.

[4] adapted this approach to a microphone setup that models the human hearing system. First the interaural time and level differences (ITD and ILD) are estimated for each source. This is done in two steps where first the ITD is estimated using the left and right ear input coherence function. Once the ITD is found the angle of arrival of the different sound sources can be computed and using the head related transfer function (HRTF) table lookup the ILD is found. Now the source parameters are known and for each source two masks are computed using two different methods (blind source separation BSS and noise-adaptive spectral magnitude expansion NASME) are constructed. At the end the masks are combined and smoothed in order to avoid artifacts.

III. ALGORITHM DESCRIPTION

As the approaches in [4] and [7] differ significantly in the details of how the masks are created it was decided to implement two simplified cocktail-party processors, one based on each paper. The main difficulty for both applications lies in the correct estimation of the source parameters as the audio recordings are non-ideal and thus correspond not anymore to the perfect setup assumption described in the papers.

A. Source separation based on [7]

In a first step, the short time Fourier transform STFT is taken over both input signals. A Hann window of length 1024 and overlap of 50% is used. The sampling rate of s_1 and s_2 is 16kHz.

Second, based on the formulas (1-4) the time-frequency parameters are computed. Each time-frequency point is labeled with these two parameters. In the third step clustering is performed with the goal to assign all time-frequency labels to n classes where n corresponds to the number of source signals. For the clustering it is assumed that the time-frequency labels are distributed according to a Gaussian mixture model and the expectation-maximization algorithm is used to classify the points.

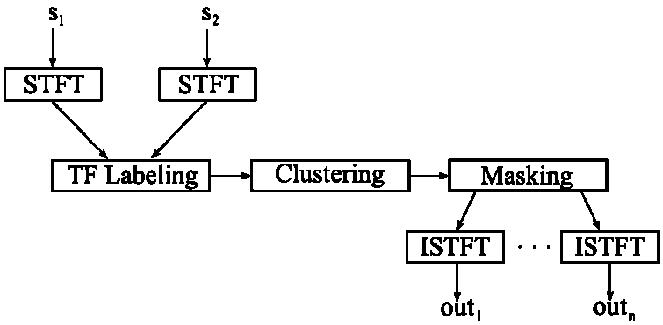

Figure 1. The block diagram of the proposed cocktail-party processor.

Once the clusters have been formed, the masks are extracted based on the cluster assignments. In order to avoid artifacts the masks are then low pass filtered using a simple 3x3 low pass filter. At the end the inverse STFT is taken and the signal is put back to the time domain. Figure 1 shows the block diagram for the implemented algorithm.

B. Source separation based on [4]

An implementation derived from [4] is pursued in this paper both for efficiency and the goals of investigating the effects of masking on the performance of source separation.

Figure 3 illustrates the block diagram of our algorithm. Firstly, the two mixed input sources are fed to the cocktail-party processor, and then we apply Short-Time Fourier Transform (STFT) with *Sin* window of size 1024 to the input signals to obtain the time-frequency maps as shown in Figure 2 for the estimation of threshold parameters (level threshold, phase threshold and masking scheme). After fixing the threshold parameters and the masking scheme, we can then generate time-frequency masks for the two mixed sources. We then mask the mixed signals with the masks to obtain the separated sources.

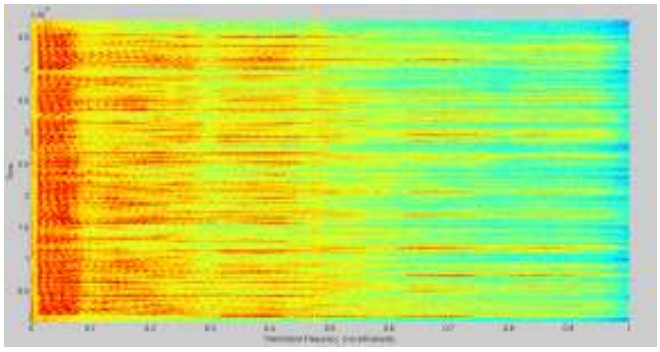

Figure 2. An example of a time-frequency map obtained by STFT, where each point on the map consists of an amplitude and a phase.

Compared with [4], our algorithm simplifies the implementation efforts in several ways: 1) use Inter-Channel Level Difference (ICLD) instead of ILD, and Inter-Channel Phase Difference (ICPD) instead of ITD, as they are easier to manipulate directly from the input signals. The definitions of these two cues used in this study are defined as follows, where amplitude and phase are obtained from each time-frequency point on the time-frequency maps of STFT as shown in Figure 2.

$$ICLD = 10\log_{10}(abs(Amplitude1/Amplitude2))$$
$$ICPD = Phase1 - Phase2$$

The crucial aspects of our algorithm are the estimation of threshold parameters and the selection of masking scheme. Presently, the estimation of threshold parameters is semi-automatic; we analyze the time-frequency maps of Inter-Channel Level Difference (ICLD), and Inter-Channel Phase Difference (ICPD) to select the thresholds based on intuition: the closer the microphone is to the source, the larger the positive ICLD, and the larger the negative ICPD for low frequency components; on the other hand, the farer the microphone from the source, the larger the negative ICLD, and the larger the positive ICPD for low frequency components. Since it is relatively unreliable for ICPD, we use the following formula to weight down its significance in the final generation of masks.

$$mask1 = 0.9 * LevelMask + 0.1 * PhaseMask$$
$$mask2 = 0.9 * LevelMask + 0.1 * PhaseMask$$

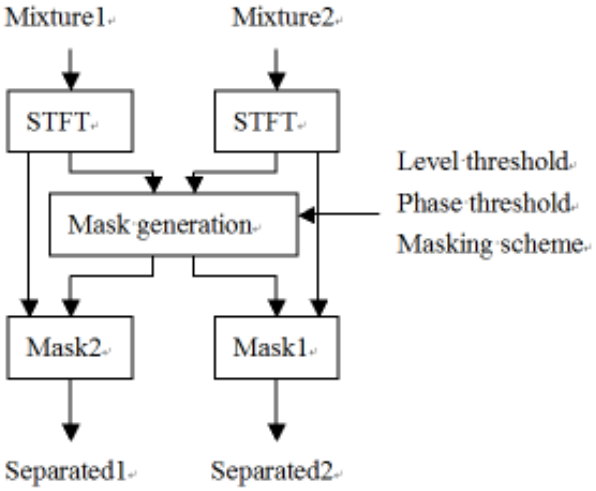

Figure 3. Block diagram of the implemented algorithm. The proposed algorithm simplifies that of [4] as our experimental setup [9] is different than that of [4], and no HRTF information is mentioned in [9]. The selection of threshold and masking scheme play key roles on the quality of source separation.

The generation of level and phase masks is based on the "W-disjoint orthogonality" as described in section II. Simply put, we estimate the likelihood of a given time-frequency point on the time-frequency map belonging to source 1 or source 2, and it is assumed that only one source dominates in a time-frequency point. The validity of this assumption will be justified in later sections.

Level masks are generated for the two sources by thresholding the ICLD, but experiments have shown that this is not enough to generate separated signals with audible quality. The tricky part is the signals with ICLD between thresholds as indicated in Figure 4. Two kinds of fitting between thresholds have been studied. We first study the performance of exponential fitting, but the overall performance is mediocre; whereas the linear fitting scheme performs better and thus becomes the choice of implementation in this paper.

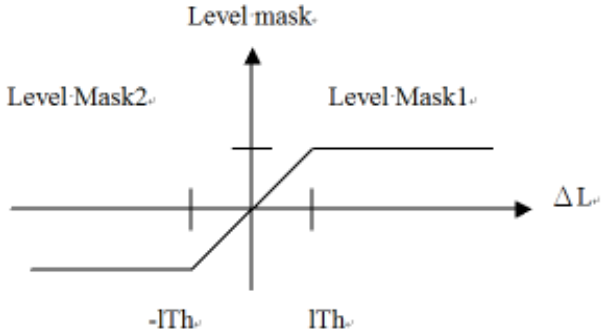

Figure 4. Level mask used in the algorithm: we threshold the inter-channel level difference in the two ends, but use linear fitting between thresholds. Level-difference thresholds are extracted from time-frequency maps, and linear fitting have been adopted for its simplicity and superior performance over other fitting schemes.

The selection of the phase masks is not as tricky as the level masks. Because phase information is periodic, we can only rely on phase information for low frequency components, and this fact reduces the overall importance of phase masks on source separation. We present this tendency by giving it a small weighting factor 0.1 as compared to a weighting factor of 0.9 for the level masks. The results of this design choice will be justified in later sections.

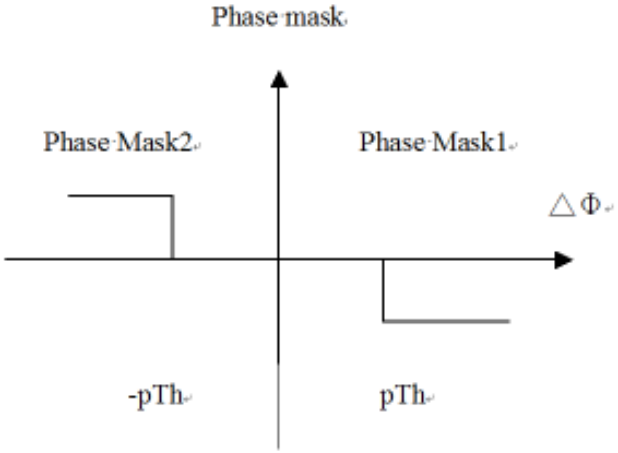

Figure 5. Phase mask used in the algorithm: we threshold the inter-channel phase difference in the two ends, and put zeros in

between. It is found in experiments that phase masks are not as reliable as level masks, so their significance weighted down in the generation of the final masks.

IV. RESULTS AND DISCUSSION

In this paper, we use three audio test sequences shown in Table 1 from [9] to experiment with our algorithms as well as different set of parameters. These test sequences are recorded in a square ordering setup with two microphones and two sound sources separated by 60 cm as shown in Figure 6 (sample rate = 16kHz). The first sequence is recorded with one human speech signal (counting numbers in English) plus music in the background; the second sequence is recorded with two human speech signals (counting numbers in English and Spanish); the third sequence is recorded with two long human speech signals (news broadcasts in English) in the noisy background. These sequences are selected for their representations of non-directional source mixture (human speech + music), directional mixture (short human speech + human speech), and noisy directional mixture (long and fast human speech + human speech in a noisy background). Clearly, for any non-directional separation algorithm, we expect it to perform better at sequence 1, but might perform badly for sequence 2 & 3, whereas a good directional separation algorithm (i.e. cocktail-party processor) should demonstrate good separation for sequence 2 & 3, and not for sequence 1. This observation will be corroborated in the later sections where we show two different algorithms derived from [7] and [4] exactly correspond to these two categories.

Sequence	Description	Setup
1	Human speech + background music	Sample rate = 16kHz
2	Simple human speech + human speech	Square ordering, sample rate = 16kHz
3	Long and fast human speech + speech in a noisy background	Square ordering, Sample rate = 16kHz

Table 1. Audio sequences, their descriptions and setup used in this paper follow that of [9].

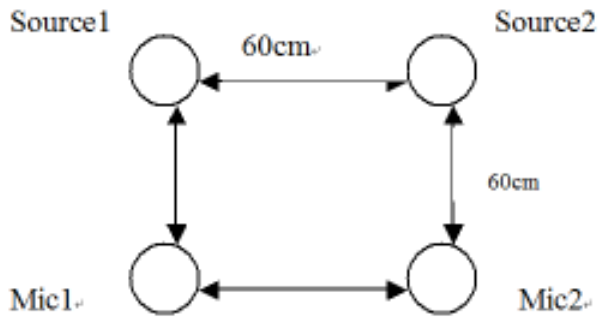

Figure 6. Experimental recording setup in [9]. Two sources and two microphones are placed in a square order with 60 cm spacing. Microphone characteristics are not mentioned in [9], and we assume no prior assumption about the characteristics of the microphone in the proposed algorithm. It is surprisingly found in this study that audible separation quality can still be obtained.

A. Source separation based on [7]

Even though the proposed method is very simple it produces very good results as long as the sources have a sufficiently large difference between their ICTD and ICLD. A very good example with nicely separated sources is a scenario where a speaker counts to ten while there is music being played at the background. If a two dimensional histogram of the ICTD and ICLD is created one can clearly see on Figure 7 that there are two different sources present. As a consequence the source separation works very well and especially for the music channel the speaker counting to ten is almost no more audible. Vice versa, for the speaker channel, the music is only faintly audible.

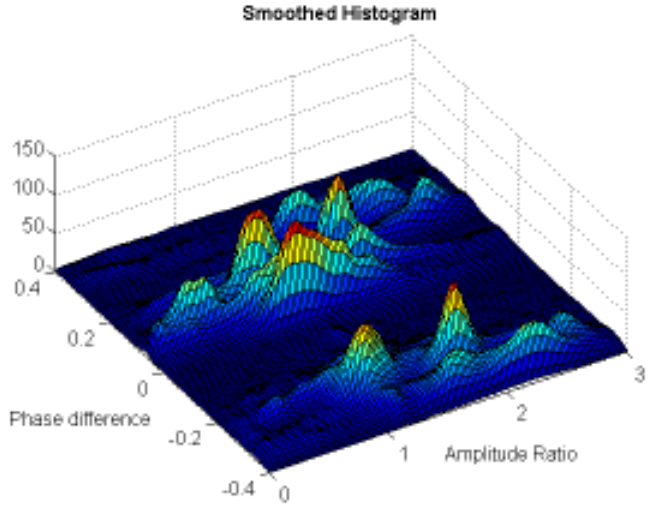

Figure 7. The histogram counting the phase differences and amplitude ratios between the two input channels. In order to get a less noisy output, the histogram has been slightly smoothed.

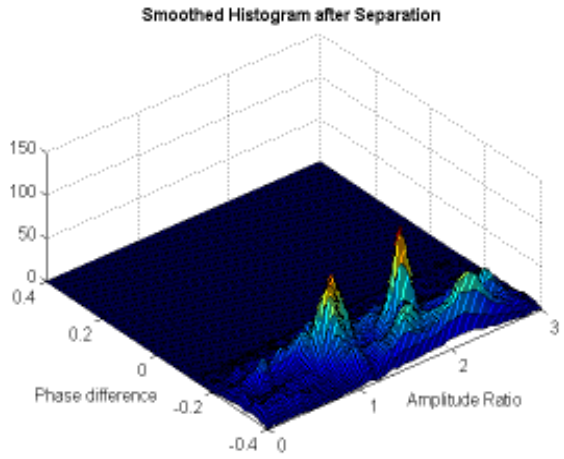

Figure 8. The same histogram computed after source separation for the source with an average phase difference of approximately 0.3. The method works very well since no points from the second source seem to be taken.

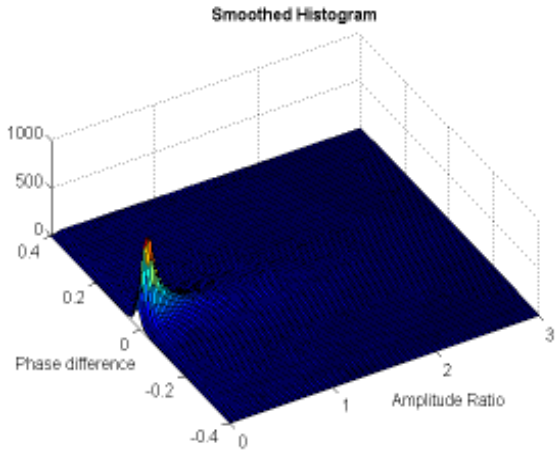

Figure 9. The histogram for a section of a concert recording where two instruments are playing. As the room impulse response contains many echoes the different ICTD and ILTDs get mixed up and become undistinguishable.

Figure 8 shows the histogram for one source after source separation. Already visually, one can see that the sources have been successfully split. On the other hand, if the sources' ICTD and ICLD lie too close to each other or even overlap, a separation with this method is no longer possible. Figure 9 illustrates the histogram from a section of a stereo concert recording where only two instruments are playing: trumpet and violin. Clearly we expect to see two peaks but there is just one. This can be explained by the fact that since it is a concert recording there are many echoes present who disturb the phase and amplitude differences for the time-frequency bins. As a consequence the algorithm does not work for these cases and a different, more sophisticated approach has to be found in order to separate the sources of such recordings, and will be shown in the next part.

The audio sequences 2 & 3 from Table 1 show a similar behavior where there is too much noise and echos in order to separate the sources successfully. The ICTD and ICLD estimates are corrupted by the noise and echos that contribute more to each time-frequency bin than the original source signal itself. However, these results could come from the limitations of [7] and other methods could provide better quality of separation as shown in the next part.

B. Source separation based on [4]

Parameters	Values	Units
Level threshold (lTh)	5	
Phase threshold (pTh)	Pi	Rad

Table 2. Level threshold and phase threshold in the experiments

We conduct experiments with audio sequences 1 ~ 3 in Table 1, and thresholds in Table 2. It is expected from Figure 3 that the proposed algorithm would work well on directional mixture, but might not be able to separate non-directional mixture. This expectation is confirmed via experimental study and the results are shown below.

We perform experiments on sequence 1 signals in Table 1 before and after separation as shown in Figure 10 & 11. We see the proposed algorithm performs poorly as the speech and background music are not separated in separated source 1, and completely rejected in separated source 2. This is reasonable as background music signals are non-directional and provide no directional cues in ICLD and ICPD.

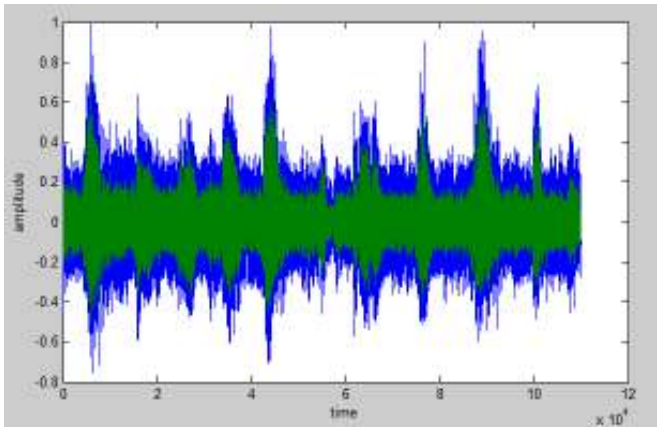

Figure 10. Source 1 signal (sequence 1) before (blue) and after (green) separation (Figure 3). We can clearly observe from the waveform of the signals that background music signals are not separated because of its non-directional essence.

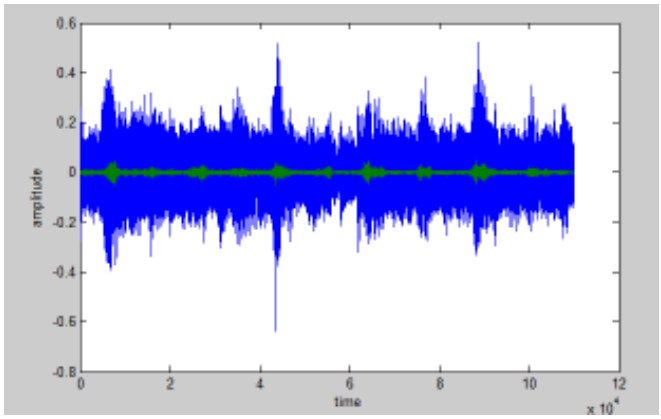

Figure 11. Source 2 signal (sequence 1) before (blue) and after (green) separation (Figure 3). We observe the signals are almost completely filtered out. This observation can be understood as the common signals (i.e. background music) are rejected completely.

We perform experiments on sequence 2 signals in Table 1 before and after separation as shown in Figure 12 & 13. The proposed algorithm works quite well for separation of speech mixture under the experimental setup in Table 1 & Figure 6. We only show the results of separation on the time domain, but it could be implicitly observed that certain kinds of separation also take place on the frequency domain. We demonstrate the time domain signals before and after separation to show the good performance of the proposed algorithms for directional speech signals. Comparing Figure 12 & 13, we see source 1 and source 2 after separation are sometimes orthogonal on the time domain – indicating the time periods where only one source dominates. This is an important characteristic of speech signals as we will also observe for the case of sequence 3.

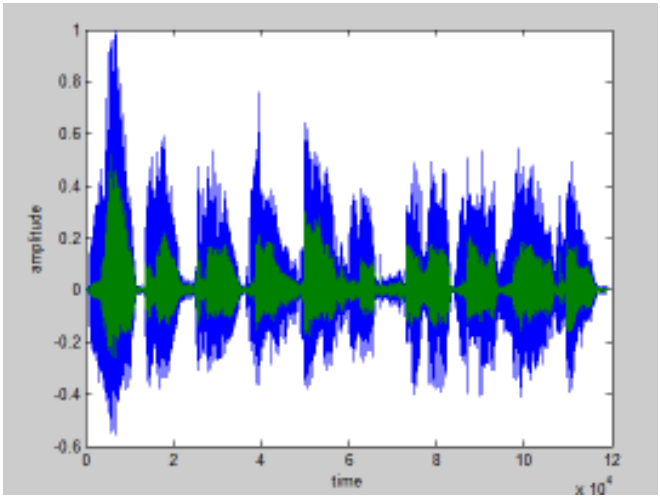

Figure 12. Source 1 signal (sequence 2) before (blue) and after (green) separation (Figure 3). We can observe some part of the speech mixture is separated on the time domain, and implicitly on the frequency domain.

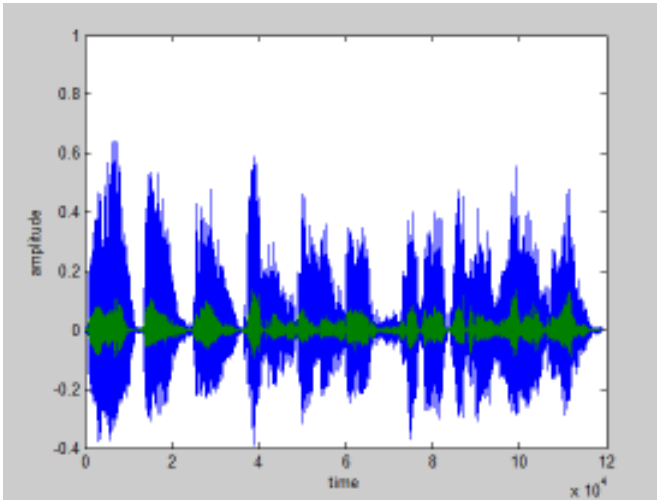

Figure 13. Source 2 signal (sequence 2) before (blue) and after (green) separation (Figure 3). We can observe some part of the speech mixture is separated on the time domain, and implicitly on the frequency domain.

We perform experiments on sequence 3 signals in Table 1 before and after separation as shown in Figure 14 & 15. Sequence 3 presents a challenge for source separation as the speech signals are long and fast varying with noisy backgrounds. Surprisingly, the proposed algorithm, thought simple compared to [4], can still perform source separation with audible quality. As we can observe, the separated signal 1 shows a clear pattern of speech signals with relatively smaller background noise, whereas the separated signal 2 shows a pattern of speech signals but with relatively larger background noise. Since we adopt no noise reduction techniques in our algorithm as compared to [4], it is surprising that we can still obtain separation with audible quality. However, as can be observed from the separated outputs, certain kinds of noise reduction techniques might help to better shape the separated signals and improve hearing quality [4].

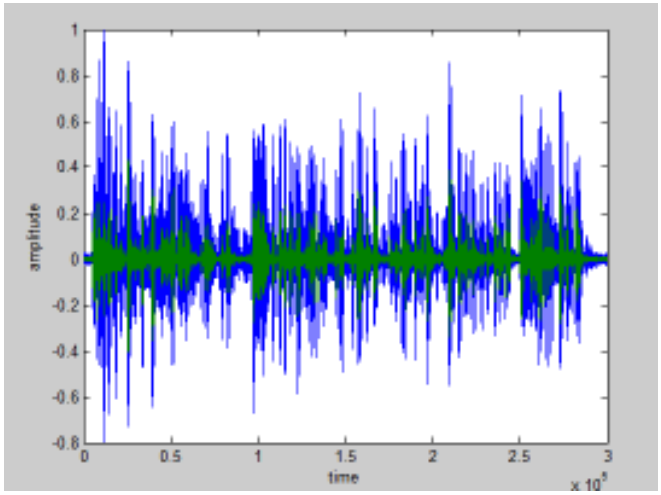

Figure 14. Source 1 signal (sequence 3) before (blue) and after (green) separation (Figure 3). We can observe some part of the speech mixture is separated on the time domain with relatively small background noise.

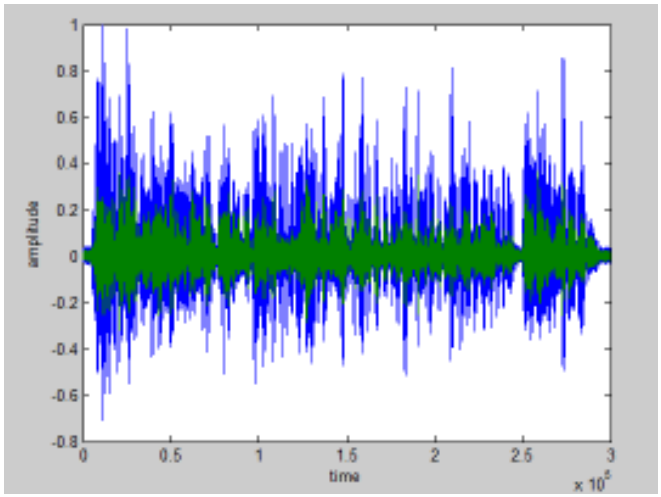

Figure 15. Source 2 signal (sequence 3) before (blue) and after (green) separation (Figure 3). We can observe some part of the speech mixture is separated on the time domain with relatively large background noise, indicating the necessity of noise reduction.

V. CONCLUSION

We present in this paper a hybrid cocktail-party processor comprising two different algorithms derived from previous published literatures [4, 7]. One algorithm works well for source separation of non-directional signals, and the other works well for directional signals (classical cocktail-party processing), but performs poorly for the non-directional one: a complementary combination.

With today's computers auditory scene analysis becomes very easy and more and more complex algorithms can be designed. The simple cocktail-party processors implemented for this project provide already a good start at source separation and they can be easily implemented on today's hardware. However, the methods are still limited in their usability and a more sophisticated scene analysis needs to be performed before

masks for source separation can be generated. The two main problems that were encountered were the noisy real-life audio files that made source parameter estimation very difficult and the constitution of the masks itself. After applying the masks, artifacts appear that would need to be filtered out (e.g. using a Wiener filter) or processed with noise reduction techniques [4].

What can also be seen from the papers are that for certain kinds of algorithm [4, 7] the source parameter estimation is a very difficult task and is still an open problem. Furthermore, the perfect cocktail-party processor that works for all different kinds of scenarios has not yet been found and is an active area of research. However, simplified algorithms such as the one presented in the second half of this paper, have shown that with simple estimation of thresholds, directional speech signals can be separated with audible quality. Given that automatic sweeping of thresholds is simple and efficient for implementation, we argue that the algorithm proposed in this paper has a better potential to realize a low-complexity cocktail-party processor.

A number of promising approaches could further enhance the work done in this paper. Some of them include 1) the efficiency of automatic sweeping of thresholds; 2) the effects of noise reduction; 3) the optimal weighting between thresholds.

Summing up, we present in this paper, a simple yet efficient realization of a cocktail-party processor via time-frequency masking. The effects of time-frequency masks are investigated in terms of histogram clustering, thresholding and linear fitting. The results show that source separation with audible quality can be obtained for non-directional speech with background music signals, and directional speech signals in both normal and difficult environments.

REFERENCES

[1] M. Bodden. Modeling human sound source localization and the cocktail-party-effect. Acta Acustica 1 , vol. 1, pp. 43–55, Ferbuary/Apr. 1993.

[2] M. Bodden, J. Blauert. Separation of concurrent speech signals: a cocktail-party-processor for speech enhancement. ETRW on speech processing in adverse conditions, Cannes-Mandelieu, France, November 1992.

[3] Harald Slaty. Algorithms for Direction specific Processing of Sound Signals - the Realization of a binaural Cocktail-Party-Processor-System. Ph.D. dissertation, Department of Electrical Engineering, Ruhr-University Bochum, 1992.

[4] Alexis Favrot, Markus Erne, Christof Faller. Improved cocktail-party processing. Proc. of the 9 th Int. Conference on Digital Audio Effects (DA Fx-06), Montreal, Canada, September 18-20, 2006.

[5] C. Faller. Parametric coding of spatial audio. Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland, July 2004, thesis No. 3062.

[6] V. R. Algazi, R. O. Duda, D. M. Thompson, and C . Avendano. The CIPIC HRTF database. Proc. IEEE Workshop App l. of Dig. Sig. Proc. to Audio and Acoust., New Palz, NY, Oct. 2001, pp. 99–102.

[7] O. Yilmaz and S. Rickard. Blind separation of speech mix-tures via time-frequency masking. IEEE Trans. Sig. Proc., vol. 52, no. 7, pp. 1830–1847, July 2004.

[8] Rasmus Kongsgaard Olsson. Algorithms for Source Separation - with Cocktail Party Applications. Informatics and Mathematical Modelling, Technical University of Denmark, IMM-PHD-2006-181.

[9] Blind Source Separation of recorded speech and music signals. http://cnl.salk.edu/~tewon/Blind/blind_audio.html